REMARKS

In the foregoing amendments, claims 2-10 and 12-25 have been amended. Claims 1-26 are currently pending, of which claims 1, 12 and 16 are independent. No new matter has been added.

I. Objections to the Specification

The Examiner objects to the disclosure because of informalities listed in the office action (office action, paragraph 2). Applicants amend the specification to address the issues raised by the Examiner. No new matter has been added by these amendments. Applicants request the Examiner to reconsider and to withdraw the objection to the specification.

The amendments are itemized below.

In the "Related Applications" section, the patent application number has been designated as "United States Patent Application No."

A duplicated period (.) at the end of the paragraph on page 2, line 2 has been removed.

The word "and" has been added after the description for Figure 3 on page 3.

A period (.) has been added after the description for Figure 4 on page 3.

All instances of the terminology "custom class" have been amended to read "custom storage class."

A white space has been added between the number "12" and the phrase "(step 62)" on page 8.

Different reference numbers have been used for user interface parameter settings and types of user interface presentation elements on pages 8 and 9.

The word "by" in the phrase "... selected parameter setting may be implemented without by the automatic code generator" has been deleted on page 9.

II. Objections to the Claims

The Examiner objects to claims 2-10, 12-15 and 17-25 because of informalities listed in the office action (office action, paragraph 3). Applicants amend claims 2-10, 12-15 and 17-25 to further clarify what the Applicants regard as the scope of the invention. Applicants request the Examiner to reconsider and to withdraw the objection to the claims.

The amendments are itemized below.

In claims 2, 3 and 18, the phrase "the code" has been replaced by "the source code."

In claims 3 and 18, the word "step" in the preamble has been replaced by "steps."

In claims 3, 13 and 18, the phrase "said custom class" has been replaced by "said custom storage class."

In claims 3, 13 and 18, a hyphen (-) has been added between the words "user" and "selected."

In claims 4, 14 and 19, the phrase "the adjusted code" has been replaced by "the adjusted source code."

In claims 4, 5, 14, 15, 19 and 20, the phrase "said view of code" has been replaced by "said view of salient aspects of the source code."

In claims 5 and 20, the phrase "segment of code" has been replaced by "segment of source code."

In claims 5-10, 13-15 and 20-25, a comma (,) has been added between the parent claim number and the word "wherein."

In claims 8 and 23, the phrase "automatically generated code" has been replaced by "automatically generated source code."

In claim 12, the white space before the semicolon (;) at the end of the first feature has been deleted.

In claims 12 and 17, the feature "code" has been replaced by "the source code."

III. Summary of Rejections

Claims 12-26 stand rejected under 35 U.S.C. §101.

Claims 1-4, 6-8, 12-14, 16-19 and 21-23 stand rejected under 35 U.S.C. §102 as being anticipated by Cheng (US 2002/0010908).

Claims 5, 15 and 20 stand rejected under 35 U.S.C. §103 as being unpatentable over Cheng in view of Childress (US 2004/0085357).

Claims 9, 10, 24 and 25 stand rejected under 35 U.S.C. §103 as being unpatentable over Cheng in view of Davidov (US 2003/0225774).

Claim 11 and 26 stand rejected under 35 U.S.C. §103 as being unpatentable over Cheng and in view of DeMaster (US 6,066,181).

These rejections will be discussed separately below.

IV. Claim Rejections under 35 U.S.C. §101

The Examiner rejects claims 12-26 under 35 U.S.C. §101 as allegedly being directed to non-statutory subject matter (office action, paragraph 5). In the foregoing claim amendments, Applicants amend independent claims 12 and 16 to further clarify the scope of the invention. Applicants respectfully request reconsideration of the outstanding rejections of claims 12-26. All claims are now believed to be in condition for allowance.

A. Claims 12-15

The Examiner rejects claims 12-15 as being "directed to systems of functional descriptive material *per se*, and hence non-statutory." In this Response, Applicants amend independent claim 12 so that it recites "an electronic device," rather than "a system" to further clarify the scope of the invention. Applicants urge reconsideration of the outstanding rejection of claim 12 under 35 U.S.C. §101.

Claims 13-15 depend from independent claim 12 and, as such, the amendments to claim 12 apply to claims 13-15. Applicants respectfully request reconsideration and allowance of claims 13-15.

B. Claims 16-26

With regard to claims 16-26, the Examiner states "Claims 16-26 recite a medium as a claimed element. However, the specification does not provide an explicit definition of what constitutes a medium. Thus, the claims can be reasonably interpreted as functional descriptive material *per se*, since the medium is not being claimed as being computer readable to provide the necessary functional and structural interrelationship to satisfy the requirements of 35 U.S.C. §101," (office action, page 5). Applicants amend independent claim 16 to refer to a "computer-readable medium" rather than a "medium" to further clarify the scope of the invention. Applicants further amend claim 16 to note that the computer-readable medium holds instructions for performing the "providing" and "creating," recited by the claim.

A computer-readable medium holding such instructions is statutory subject matter. See *In re Beauregard*, 35 USPQ.2d 1383 (Fed. Cir. 1995). The Examiner agrees with this position at page 5 of the office action: "a claimed computer readable medium encoded with a computer program is a computer element, which defines structural and functional interrelationships between the computer program and the rest of the computer, that permits the computer program's functionalities, and is thus statutory." Accordingly, Applicants urge reconsideration of the outstanding rejection of claim 16.

Claims 17-26 depend from independent claim 16 and are patentable for at least the same reasons as claim 16. Applicants respectfully request reconsideration and allowance of claims 17-26.

V. Claim Rejections under 35 U.S.C. §102

The Examiner rejects claims 1-4, 6-8, 12-14, 16-19 and 21-23 under 35 U.S.C. §102(b) as being anticipated by Cheng (US 2002/0010908) (office action, paragraph 7). Applicants respectfully traverse the rejection as set forth below.

A. Claim 1

Independent claim 1 recites:

> "In an electronic device having a graphical modeling and execution environment, said graphical modeling and execution environment including at least one graphical model, a method comprising the steps of:
>
> providing a user interface with a plurality of selectable parameters for a custom storage class, said custom storage class specifying the manner in which an automatic code generator creates source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment; and
>
> creating a custom storage class in said graphical modeling and execution environment utilizing parameters selected by a user from said plurality of selectable parameters."

Cheng discusses command tree 1, which is a hierarchical representation of the commands available to a user of an operating system. When a user enters a command on the command line interface (CLI), the operating system parses the command and determines the appropriate action to take. The operating system then traverses command tree 1 until the operating system reaches the appropriate leaf of the tree. See Cheng, paragraph [0018]. The system of Cheng allows a developer to quickly create CLI commands by creating a structure of command nodes. Each command node may, in turn, be edited to add parameters and handler functions. See Cheng, paragraph [0019].

Cheng includes a code generating engine for automatically generating software code for a handler function using handler function information and/or parameter information. Cheng provides a graphical user interface for receiving parameter information and handler function information. See Cheng, paragraph [0005].

Applicants respectfully submit that Cheng fails to disclose at least the following feature of claim 1: "a custom storage class, said custom storage class specifying the manner in which an automatic code generator creates source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment." As will be explained in more detail below, various aspects of this feature are not disclosed by Cheng.

The Examiner points to the following language (office action, pages 7 and 9-10) at paragraph [0028]:

> The entering of parameters is also accomplished via GUI by adding the desired parameters to parameter field.

The Examiner lastly points to the following language (see office action, pages 7 and 9-10) at paragraph [0036]:

> ... the command structure is generated by command structure generation engine using the information entered by the developer.

The Examiner also points to the following language (see office action, pages 7 and 9-10) at paragraph [0043] as disclosing the feature:

> ... the handler function code is generated by handler code generation engine. As described above, the handler function has associated software code.... Handler code generation engine automatically generates this software code using the information entered by the developer and the parameter and handler function definitions generated by command structure generation engine.

The above excerpts discuss that a user may enter parameter information via a GUI, and that a command structure generation engine generates parameter and handler function definitions. Information entered by the user and the parameter and handler function definitions are then used to generate software code associated with a handler function. As will be explained in more detail below, these three cited sections do not disclose the quoted feature of claim 1.

First, Cheng fails to disclose a "custom storage class," as recited by claim 1. A custom storage class specifies the manner in which an automatic code generator creates source code for data referenced by a graphical model. The custom storage class recited by claim 1 is a class, which is an object oriented feature. Cheng does not teach or suggest using a custom storage class. Further, the parameter and handler function definition files, as recited in Cheng, are not shown to be object-oriented features.

Second, Cheng does not disclose "specifying the manner in which an automatic code generator creates source code," as required by claim 1. From the above-quoted language, it appears that the Examiner is pointing to the parameter definition files and handler function definition files as satisfying this claim language. However, the parameter definition files and the handler function definition files do not specify the manner in which the automatic code generator creates source code. Cheng states: "Command structure generation engine 145 takes the information entered by the developer and generates the handler function definition file. The exemplary code above may include the following information: the type of command (e.g., can this command handle "No" forms), the bitmask of required parameters, the bitmask of optional parameters and the actual handler function associated with the definition." See Cheng, paragraph [0042]. As described in the above excerpt, in Cheng, the parameter definition file and handler function definition file generated by the command structure generation engine include information on the parameter function and handler function, respectively. In contrast, the custom storage class recited by claim 1 specifies the manner in which an automatic code generator creates source code. Cheng does not disclose that the definition files specify "the manner in which an automatic code generator creates source code," as required by claim 1.

Third, Cheng does not disclose "source code corresponding to *data referenced by said graphical model* in said graphical modeling and execution environment," as required by claim 1. From the above-quoted language cited by the Examiner, it appears that the Examiner is pointing to the handler function as satisfying this claim language. However, code generated in Cheng does not correspond to data referenced by a graphical model. Cheng states: "Command tree 1 is a hierarchical representation of the commands available to the user of the operating system." See Cheng, paragraph [0018]. Cheng further states: "Each of command nodes 155-180 has either an associated handler function or an associated parameter." See Cheng, paragraph [0022]. Cheng discusses that the parameters and handler functions are associated with command nodes in a command tree, the command tree being a representation of the commands available to the user of the operating system. Cheng further states: "the handler function has associated software code." See Cheng, paragraph [0043] (cited by the Examiner). The software code in Cheng is associated with the handler function. The handler function is, in turn, associated with the command nodes in a command tree. Thus, in Cheng, the software code corresponds to *command nodes in a command tree*. In contrast, claim 1 of the current application requires that the source code corresponds to *data referenced by a graphical model in a graphical modeling and execution environment*. Thus, Cheng fails to disclose that the software code associated with the handler function "corresponds to data referenced by a graphical model," as required by claim 1.

In view of the above arguments to claim 1, Applicants respectfully request reconsideration and allowance of claim 1.

B. Claims 2-4 and 6-8

Claims 2-4 and 6-8 depend directly or indirectly from independent claim 1 and, as such, incorporate all of the features of claim 1. Accordingly, claims 2-4 and 6-8 are allowable for at least the reasons set forth above with respect to claim 1. Applicants respectfully request reconsideration and allowance of claims 2-4 and 6-8.

C. Claims 12 and 16

Amended independent claim 12 recites:

> "An electronic device having a modeling and execution environment with at least one graphical model, said electronic device comprising:
>
> a user interface with a plurality of selectable parameters for a custom storage class, said custom storage class specifying the manner in which an automatic code generator creates source code from said graphical model;
>
> a custom storage class in said graphical modeling and execution environment, said custom storage class created utilizing parameters selected by a user from said plurality of selectable parameters; and
>
> a view of salient aspects of the source code generated by said automatic code generator utilizing the user-selected parameters."

Amended independent claim 16 recites:

> "A computer-readable medium for use in an electronic device having a graphical modeling and execution environment, said graphical modeling and execution environment including at least one graphical model, said medium holding instructions for:
>
> providing a user interface with a plurality of selectable parameters for a custom storage class, said custom storage class specifying the manner in which an automatic code generator creates source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment; and
>
> creating a custom storage class in said graphical modeling and execution environment utilizing parameters selected by a user from said plurality of selectable parameters."

Applicants respectfully submit that Cheng fails to disclose at least the following feature of claims 12 and 16: "a custom storage class, said custom storage class specifying the manner in which an automatic code generator creates source code" from said graphical model as discussed above in connection with claim 1, and therefore claims 12 and 16 are patentable for at least this reason. Applicants respectfully request reconsideration and allowance of claims 12 and 16.

D. Claims 13-14, 17-19 and 21-23

Claims 13 and 14 depend from independent claim 12 and, as such, are allowable for at least the reasons set forth above with respect to claim 12. Applicants respectfully request reconsideration and allowance of claims 13 and 14.

Claims 17-19 and 21-23 depend from independent claim 16 and, as such are allowable for at least the reasons set forth above with respect to claim 16. Applicants respectfully request reconsideration and allowance of claims 17-19 and 21-23.

VI. Claim Rejections under 35 U.S.C. §103

A. Claims 5, 15 and 20

The Examiner rejects claims 5, 15 and 20 under 35 U.S.C. §103(a) as being unpatentable over Cheng in view of Childress (office action, paragraph 9). Applicants respectfully traverse the rejections for the reasons set forth below.

Cheng has been summarized above.

Childress is generally directed to providing systems and methods for creating, viewing and/or modifying business rules used by an automated insurance claim processing system. A rule editor may provide a user with a graphical display of at least a portion of a business rule implemented in software. See Childress, paragraphs [0009-0010]. Business rules of knowledge-based system encode the formulas used in evaluating insurance claims in an insurance claim processing software. See Childress, paragraph [0007].

I) Claim 5

Claim 5 depends on independent claim 1. Applicants respectfully submit that Cheng and Childress, alone or in combination, fail to teach or suggest all of the features of claim 5.

Applicants respectfully submit that Cheng and Childress fail to teach or suggest at least the following feature of claim 5: "a custom storage class, said custom storage class specifying the manner in which an automatic code generator creates source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment."

Cheng has been discussed extensively above. Applicants respectfully submit that Cheng fails to teach or suggest all of the features of claim 5. The teachings of Childress do not supplement Cheng in such a way as to cure the failure to recite the custom storage class with claimed limitations.

Childress is directed to generating code corresponding to *business rule components in the form of entries in a database table*. In contrast, claims 1 and 5 both recite "specifying the manner in which an automatic code generator creates source code corresponding to data referenced by said graphical model." Claim 5 depends on claim 1 and is therefore not rendered obvious by the combination of Cheng and Childress because Childress does not cure the deficiency left by Cheng with regard to the aforementioned feature of claim 1.

ii) Claims 15 and 20

Claims 15 and 20 depend on independent claims 12 and 16, respectively. Applicants respectfully submit that Cheng and Childress, alone or in combination, fail to teach or suggest all of the features of claims 15 and 20.

Applicants respectfully submit that Cheng and Childress fail to teach or suggest at least the following feature of claims 15 and 20: "a custom storage class, said custom storage class specifying the manner in which an automatic code generator creates source code" from a graphical model.

Cheng has been discussed extensively above. Applicants respectfully submit that Cheng fails to teach or suggest all of the features of claims 15 and 20. The teachings of Childress do not supplement Cheng in such a way as to cure the failure to recite the custom storage class with claimed limitations.

Childress is directed to generating code from *business rule components.* In contrast, claims 12, 15, 16 and 20 all recite "specifying the manner in which an automatic creates source code" from a graphical model. Claim 15 depends on claim 12 and is therefore not rendered obvious by the combination of Cheng and Childress because Childress does not cure the deficiency left by Cheng with regard to the aforementioned feature of claim 12. Claim 20 depends on claim 16 and is therefore not rendered obvious by the combination of Cheng and Childress because Childress does not cure the deficiency left by Cheng with regard to the aforementioned feature of claim 16.

B. Claims 9-10 and 24-25

The Examiner rejects claims 9, 10, 24, and 25 under 35 U.S.C. §103(a) as being unpatentable over Cheng in view of Davidov (office action, paragraph 10). Applicants respectfully traverse the rejection for the reasons set forth below.

Cheng has been summarized above.

Davidov is generally directed to an infrastructure for creating applications for mobile information devices, using a tag-based markup language. Developers can use the markup language to define applications and content based on easily manipulated textual tags, rather than having to write specific code. A compiler, which is optionally controllable via a command line interface, processes the tags in several phases. Initially, the input is parsed in order to check for errors. Next, a hierarchical object model of the application is populated with objects corresponding to the tags. Then, source code files are generated, which include supporting resource files (such as images) corresponding to the objects in the hierarchy. In producing the source code files, a generator traverses the object model hierarchy in a top-down manner, producing a source file for different elements, for example, screens, forms, and servlets. For each element required to be generated, an appropriately configured generator class is invoked in order to generate appropriate code from the data stored in the object model. See Davidov, paragraph [0013].

i) Claims 9 and 10

Claims 9 and 10 depend on independent claim 1. Applicants respectfully submit that Cheng and Davidov do not teach or suggest all of the features of claims 9 and 10.

Applicants respectfully submit that Cheng and Davidov fail to teach or suggest at least the following feature of claims 9 and 10: "a custom storage class, said custom storage class specifying the manner in which an automatic code generator creates source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment."

Cheng has been discussed extensively above. Applicants respectfully submit that Cheng fails to teach or suggest all of the features of claims 9 and 10. The teachings of Davidov do not supplement Cheng in such a way as to cure the failure to recite the custom storage class with claimed limitations.

Davidov is directed to generating source code from an object model hierarchy including screens, forms and servlets. In contrast, claims 1, 9 and 10 all recite "specifying the manner in which an automatic code generator creates source code corresponding to data referenced by said graphical model." Claims 9 and 10 depend on claim 1 and are therefore not rendered obvious by the combination of Cheng and Davidov because Davidov does not cure the deficiency left by Cheng with regard to the aforementioned feature of claim 1.

ii) Claims 24 and 25

Claims 24 and 25 depend on independent claim 16 and, as such, incorporate all of the features of claim 16. Applicants respectfully submit that Cheng and Davidov, alone or in combination, fail to teach or suggest all of the features of claims 24 and 25.

Applicants respectfully submit that Cheng and Davidov do not teach or suggest at least the following feature of claims 24 and 25: "specifying the manner in which an automatic code generator creates source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment."

Cheng has been discussed extensively above. Applicants respectfully submit that Cheng fails to teach or suggest all of the features of claims 24 and 25. The teachings of Davidov do not supplement Cheng in such a way as to cure the failure to recite the aforementioned teaching of claims 24 and 25.

Davidov is directed to generating source code from an object model hierarchy including screens, forms and servlets. In contrast, claims 16, 24 and 25 all recite "specifying the manner in which an automatic code generator creates source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment." Claims 24 and 25 depend on claim 16 and are therefore not rendered obvious by the combination of Cheng and Davidov because Davidov does not cure the deficiency left by Cheng with regard to the aforementioned feature of claim 16.

C. Claims 11 and 26

The Examiner rejects claims 11 and 26 under 35 U.S.C. 103(a) as being unpatentable over Cheng in view of DeMaster (office action, paragraph 11). Applicants respectfully traverse the rejection for the reasons set forth below.

Cheng has been summarized above.

DeMaster is generally directed a Java native interface code generator to facilitate mixed language programming. The Java native interface code generator makes native code programmed in a native language, such as C, C++ or Assembly, accessible to Java application programs. A programmer initially generates a native interface definition for native code, preferably using a neutral Interface Definition Language (IDL) to describe native code components, namely, the functions, data structures, constants and other user-defined data types, for which native mappings are to be generated. The Java native interface code generator also supports complex data types, including nested data structures and multi-dimensional arrays (DeMaster, column 2, lines 5-17).

i) Claim 11

Claim 11 depends on independent claim 1 and, as such, incorporates all of the features of claim 1. Applicants respectfully submit that Cheng and DeMaster do not teach or suggest all of the features of claim 11.

Applicants respectfully submit that Cheng and DeMaster fail to teach or suggest at least the following feature of claim 11: "specifying the manner in which an automatic code generator creates source code corresponding to data referenced by said graphical model."

Cheng has been discussed extensively above. Applicants respectfully submit that Cheng fails to teach or suggest all of the features of claim 11. The teachings of DeMaster do not supplement Cheng in such a way as to cure the failure to recite the above feature of claim 11.

DeMaster is directed to generating code corresponding to *user-derived native interface definition*. In contrast, claims 1 and 11 both recite "specifying the manner in which an automatic code generator creates source code corresponding to data referenced by said graphical model." Claim 11 depends on claim 1 and is therefore not rendered obvious by the combination of Cheng and DeMaster because DeMaster does not cure the deficiency left by Cheng with regard to the aforementioned feature of claim 1.

ii) Claim 26

Claim 26 depends on independent claim 16. Applicants respectfully submit that Cheng and DeMaster, alone or in combination, do not teach or suggest all of the features of claim 26.

Applicants respectfully submit that Cheng and DeMaster fail to teach or suggest at least the following feature of claim 26: "specifying the manner in which an automatic code generator creates source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment."

Cheng has been discussed extensively above. Applicants respectfully submit that Cheng fails to teach or suggest all of the features of claim 26 and DeMaster does not supplement the teachings of Cheng in such a way as to cure this deficiency.

DeMaster is directed to generating code from *native interface definitions*. In contrast, claims 16 and 26 both recite "specifying the manner in which an automatic code generator creates source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment." Claim 26 depends on claim 16 and is therefore not rendered obvious by the combination of Cheng and DeMaster because DeMaster does not cure the deficiency left by Cheng with regard to the aforementioned feature of claim 16.

CONCLUSION

In light of the foregoing arguments, Applicants respectfully submit that Cheng, Childress, Davidov and DeMaster, alone or in any reasonable combination, fail to disclose, teach or suggest the patentable features of the invention, and contends that the claimed invention is novel and non-obvious in view of the aforementioned references.

Please charge any shortage or credit any overpayment of fees to our Deposit Account No. 12-0080, under Order No. MWS-062. In the event that a petition for an extension of time is required to be submitted herewith, and the requisite petition does not accompany this response, the undersigned hereby petitions under 37 C.F.R. §1.136(a) for an extension of time for as many months as are required to render this submission timely. Any fee due is authorized to be charged to the aforementioned Deposit Account.

In view of the above comments, Applicants believe that the pending application is in condition for allowance and urge the Examiner to pass the claims to allowance. Should the Examiner feel that a teleconference would expedite the prosecution of this application, the Examiner is urged to contact the Applicants' attorney at (617) 227-7400.

Dated: January 22, 2007

Respectfully submitted,

By Kevin J. Canning
Kevin J. Canning
Registration No.: 35,470
LAHIVE & COCKFIELD, LLP
One Post Office Square
Boston, Massachusetts 02109-2127
(617) 227-7400
(617) 742-4214 (Fax)
Attorney/Agent For Applicant